Raptor – Software and Applications on BlueGene/L*

Jeff Greenough, Allen Kuhl, Louis Howell, Alek Shestakov,
Ulrike Creach, Al Miller and Ellen Tarwater
Andrew Cook and Bill Cabot

AX-Division, LLNL

BlueGene/L: Applications, Architecture and Software Workshop
October 14-15, 2003
Reno, NV

*This work was performed under the auspices of the US Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No.W-7405-ENG-48.

Outline

- Project and Software Overview
- Current work/performance on MCR & ALC
- Mapping to BG/L
- Applications on BG/L

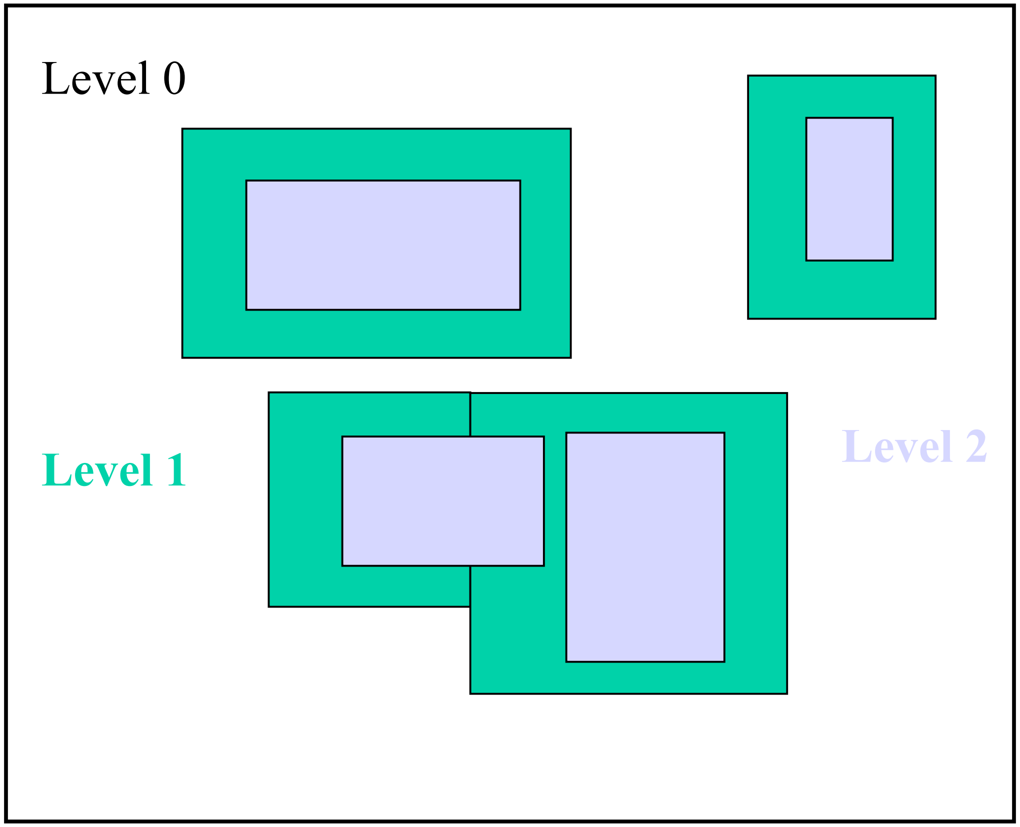
Project Overview

- AMRh is the code development project; *Raptor* is the code.
  - Develop modern numerical methods in a software framework that supports Adaptive Mesh Refinement (AMR).
  - Apply the high-resolution numerical methods to problems related to instability development and transition to turbulence → closely coupled to the Complex Hydrodynamics Program (P. Miller)
    - The goal is to use Raptor to provide detailed simulation data and augment experimental data for guiding/verifying model development. Efficiently carry out parameter/sensitivity studies at high resolution.
    - Use as a testbed for developing a high resolution LES capability for shock-driven flows. Examine underlying Godunov integrator, investigate scheme improvements, utility of SGS models, etc.

AMR Overview

- AMR is a technique for efficiently increasing resolution in space and time.
- Maximize resolution for a fixed computational cost.

- Base Grid is Level 0 and covers all of computational domain
- Locally refine spatially to create a new level. Tag structures, errors, etc. for refinement. Create blocks.
- Finer levels strictly contained in next coarser level -> proper nesting
- Sub-cycle finer grids in time (temporal resolution)
- Data on a level (union of grids) maintained as a fundamental object



AMR Utility

• AMR is an enabling technology for grid refinement, parameter and sensitivity studies. Making this the rule rather than the exception.

M=1.2 RM Initial Amplitude Sensitivity Study

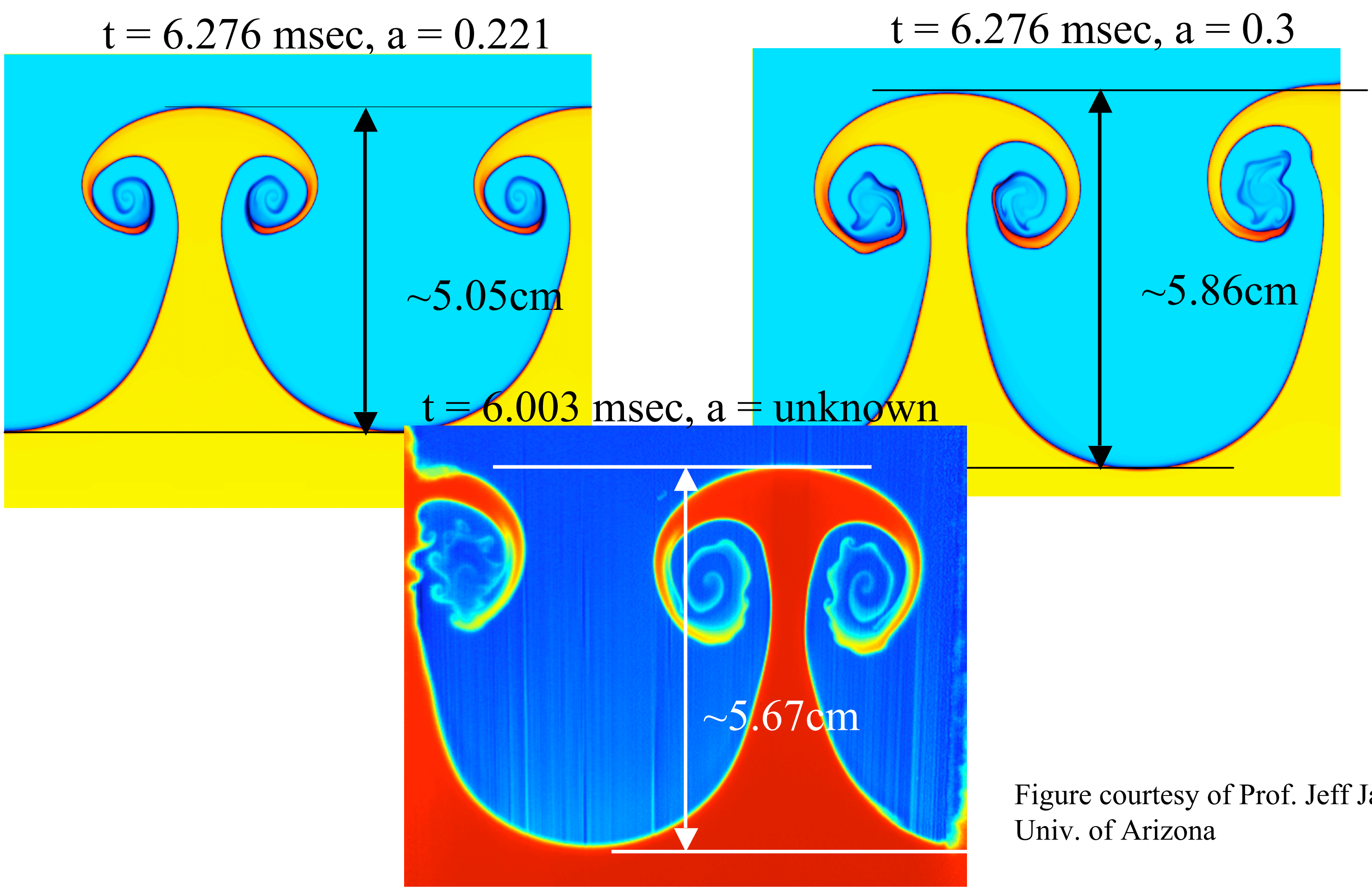


Figure courtesy of Prof. Jeff Jacobs, Univ. of Arizona

Godunov Method Improvements

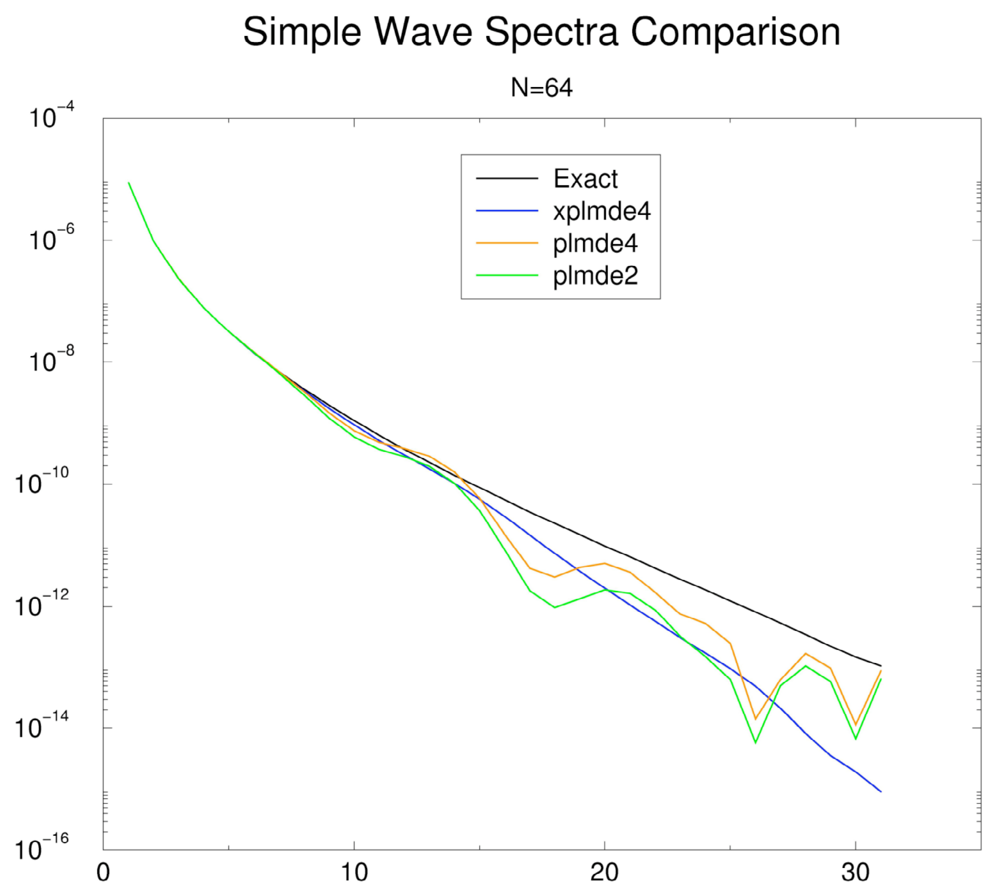
• Modern Godunov method seems to be a good second order scheme for shock-driven flows (P. Colella, 1985).
• Pursuing improvements to the current method to give a good match for LES.

• Simple Wave is an exact solution to the Euler Eqn's before breaking.

• Examine the spectra at $t=0.75\ t_b$.

• Compare exact solution spectra to result using different slope limiters.

• Follow "best practice" of LES SGS modelling community.



Software Overview

- Raptor is a hybrid C++/Fortran code
  - Current code is based on BoxLib infrastructure, provided by CCSE (John Bell) at LBNL, (http://seesar.lbl.gov/ccse/Software/#anchor-apps) which is part of a larger applications suite.
    - Base Libraries provide data container classes, high level memory management, runtime profiling, interface to I/O routines and parallelism.
      - Parallelization models: MPI and pthreads.
        - Point-to-point communication is asynchronous. Global reduction at the end of a timestep.
        - Each processor communicates with any other processor only once. Efficient use of buffers.
        - Grids on a level is the fundamental object. Distribute Grids among the available processors.

Software Overview

- I/O model: N processes write to N files.
    - Explicit barrier before writing restart/visualization files.
    - Number of CPU's before/after restart can be different.

- Supporting libraries

    - AmrLib is an extension of BoxLib supporting adaptive, block, datasets (hierarchical data and data on a level). Virtual base classes.

        - AMR case → multiple levels of data, multiple grids.

        - non-AMR case → single level of data, multiple grids.

    - BndryLib is also provided which gives boundary object support (physical boundary conditions, geometrical support and coarse/fine grid operations).

Software Overview

• LLNL owns the physics modules and associated multi-level objects (instantiation of the virtual base class).

- Explicit methods (multifluid hydrodynamics)
    - Conservative, upwind, monotone finite difference methods (Godunov).
    - Nearest neighbor algorithm using ghost cells (3).

- Implicit methods (diffusion-radiation)
    - Requires solving a multi-block banded linear system.
    - Requires 1 ghost cell of Dirichlet data.
    - Currently reliant on *Hypre* package (multigrid based methods).

Software Overview

• Coarse/fine level synchronization algorithms (refluxing) are required for global, with respect to the hierarchy, conservation.

- Explicit methods use explicit refluxing
  - Special purpose boundary object accumulate flux mismatch. Coarse level updated.

- Implicit methods use implicit synchronization
  - Flux mismatch accumulated. Both fine and coarse levels updated iteratively.

Software Overview

- Current code builds and runs on LC Linux clusters (IA32 system).
    - Running on other LC platforms as well (gps, tc2k, frost, etc.)

- Uses g++ and g77. GNUmake file system.
    - Intel and IBM compilers also.

- Additional libraries for *pdb* support
    - Tabular EOS and opacity database format (LEOS). Data is exported controlled.

Code Capabilities

- Physics capabilities (working)
  - Full set of coordinate systems
    - Cartesian, cylindrical and spherical in 1,2,3D)
  - Hydrodynamics (multi-fluid capturing or tracking)
  - Gray Radiation diffusion
  - Electron conduction
  - Analytic/tabular EOS/opacity (export controlled data)

- Physics capabilities (under development)
  - Multi-group diffusion

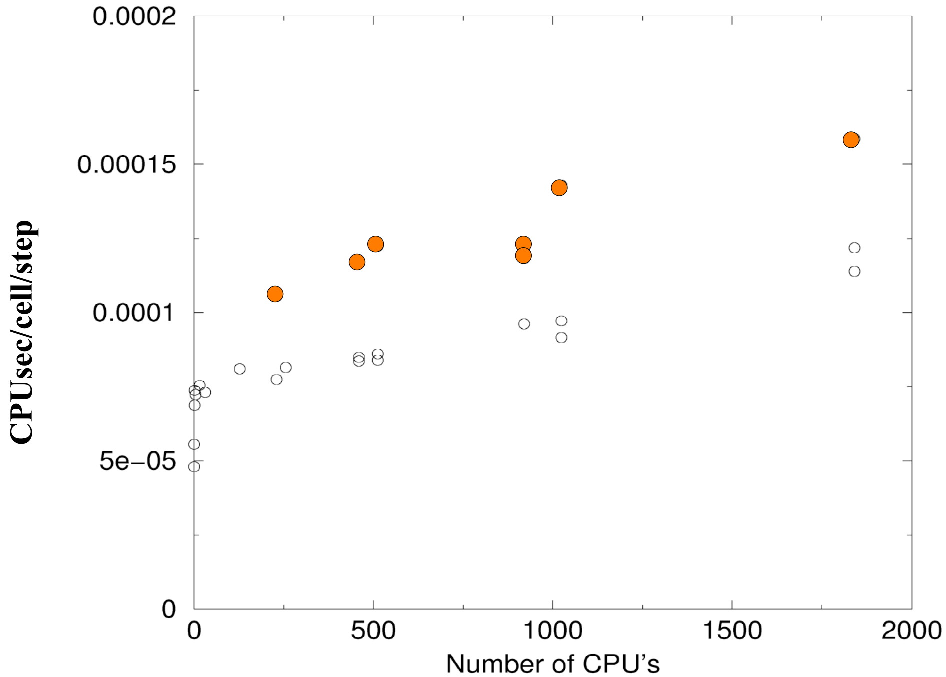
- C-preprocessor directives control physics package inclusion and Makefile controls linked libraries.

Current Work on ALC & MCR

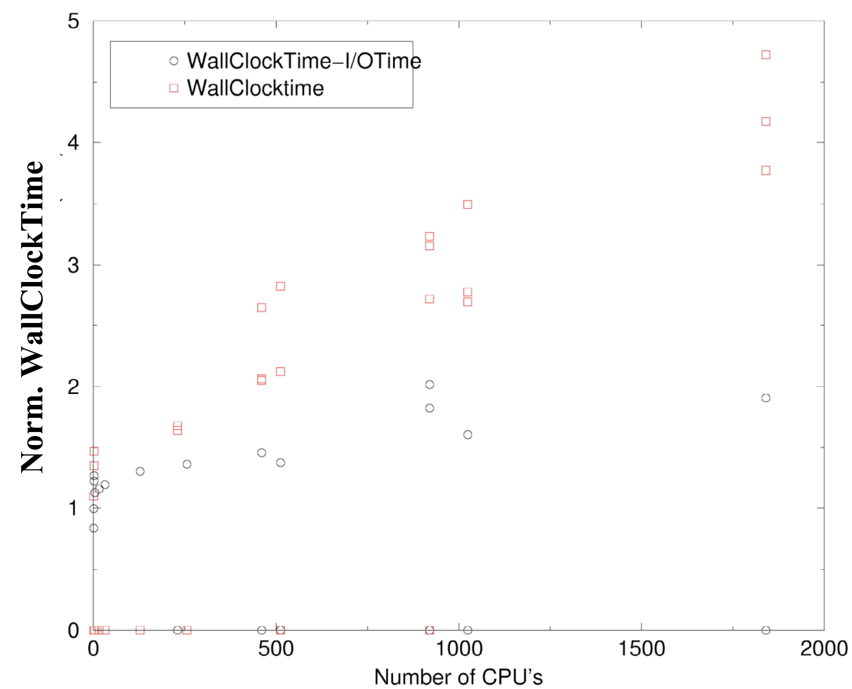
- Raptor has been running at scale on ALC & MCR since July ’03,
    - Supporting RT simulations of A. Cook and W. Cabot, LLNL.
    - Performing high-resolution 3D multi-mode RM simulations.

- The test problem is a 3D single level (multiple grids) RT 2-fluid problem (hydrodynamics only).
    - L x L x 2L. Periodic transverse boundary condition. Rigid top and bottom boundaries.
    - One 32^3 grid per CPU, i.e. workload per CPU is fixed.
    - Block size is runtime adjustable. 32^3 is cache friendly; surface to volume ratio is about 1/2.

Scaling on ALC & MCR



Unset envirn. variables to get > 2Gb memory

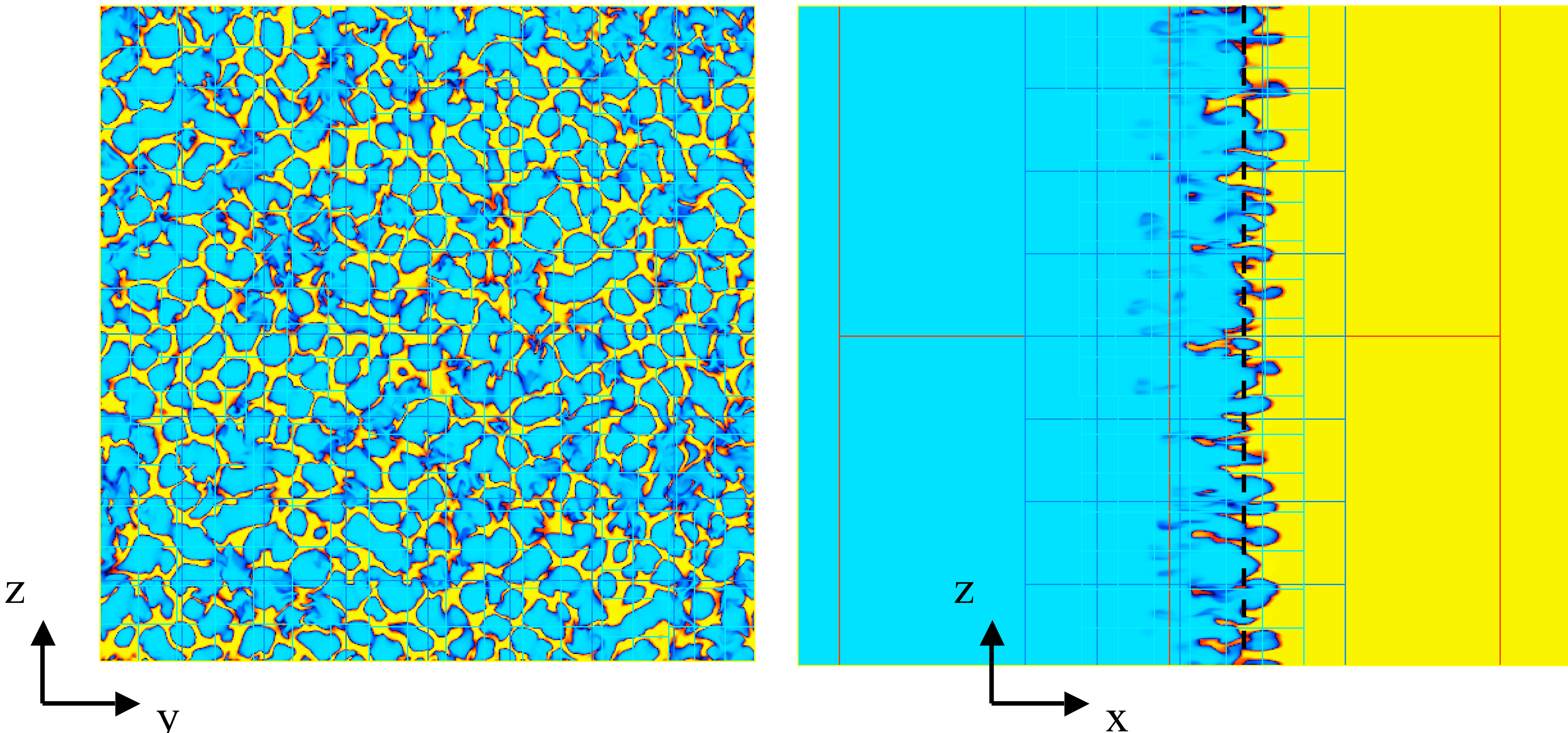


- g++/f77 results
- About 94% overall parallel efficiency
- Wall clock time is relatively flat

Current Work on ALC & MCR

- 3D multimode RM. M=1.3, At=0.6 (Air/Sf_6 model).
    - Lab-fixed coordinates. 1.5m x 10cm x 10cm. Rigid endwall.
    - 512^2 effective transverse resolution, $\Delta x = 195 \mu m$

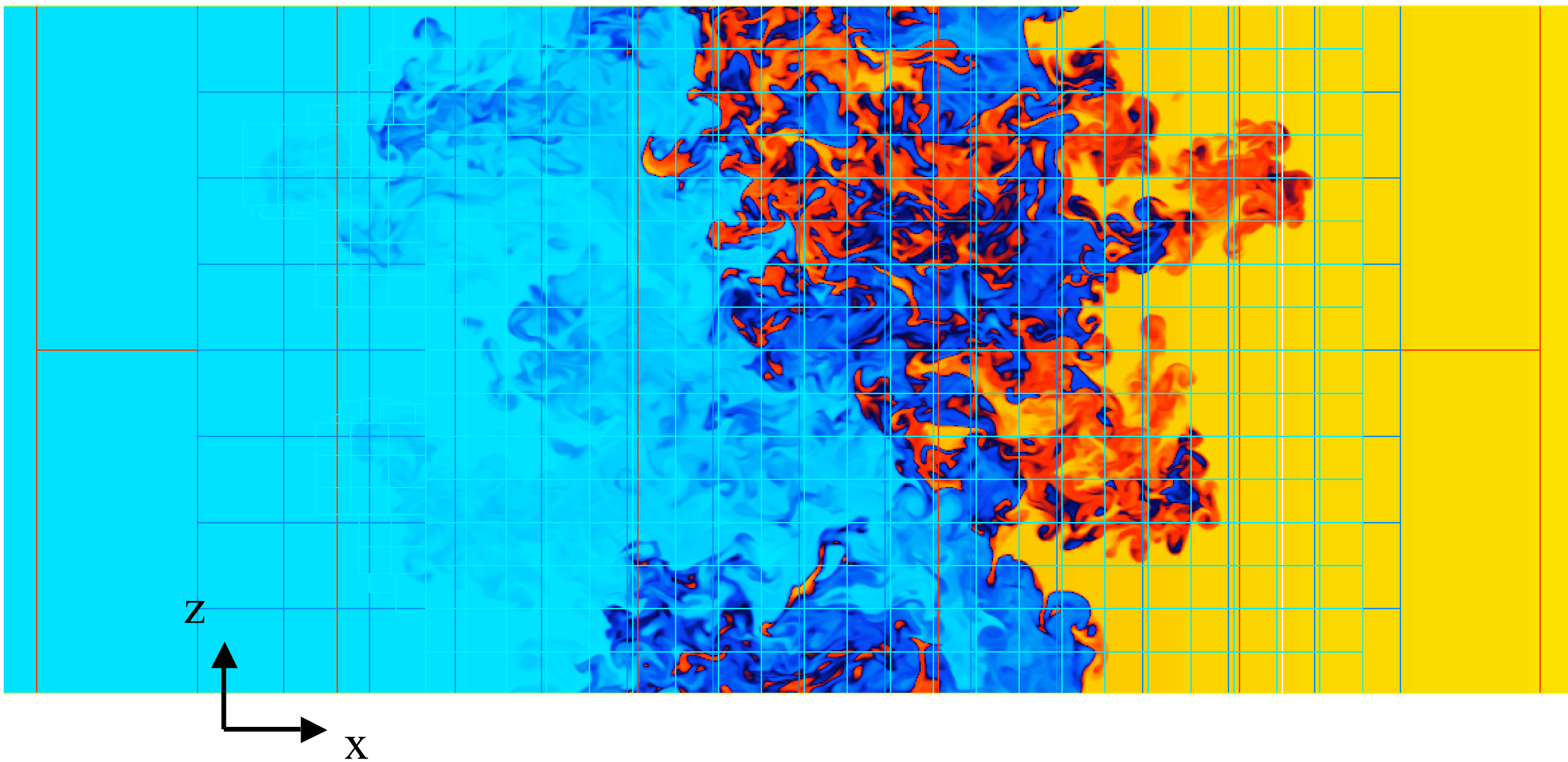
View of density field – Early time after first shock



Current Work on ALC & MCR

- 3D multimode RM. M=1.3, At=0.6 (Air/Sf_6 model)
    - 512^2 effective transverse resolution

View of density field slice – Late time after multiple accelerations



Current Work on ALC & MCR

- Final statistics from AMR run:

    - 8.2 days of CPU time. 2.5@1920 CPU's + 5.7@1024 CPU's
        - Simulation time of 23 msec

    - 205M cells on finest level → ~ 7700 32^3 grids
        - At t=0, 13M cells → ~ 512 32^3 grids

        - Could re-tune problem by increasing block size. Reduce the block count (same cell count). Reduce communication, reduce surface to volume ratio. Performance effect?

Mapping to BG/L

• Currently require about 30Mb per 32^3 grid block for a 2 fluid prototype problem (targeting RM/RT).

• 32Tb available on 65536 compute nodes of BG/L gives ***3270^3 effective problem size***.

• Still room for improvement:

  • Only one time level of data required for finest level (or single level). Near factor of two decrease in memory requirement. ***Effective problem size is 4030^3***

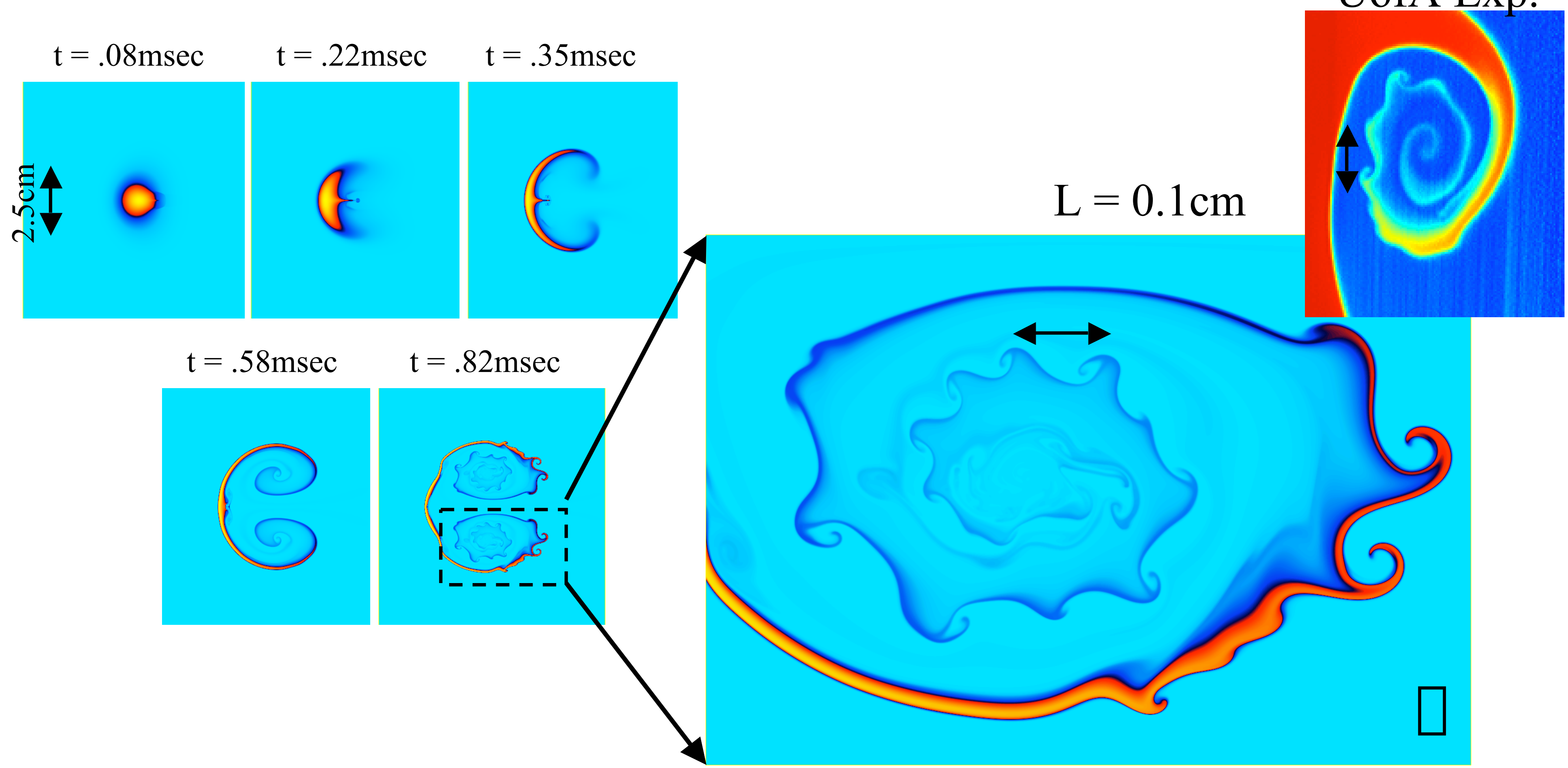
Mapping to BG/L

- Currently grid distribution algorithms do not consider physical proximity.
    - Infrastructure support for modifying grid distribution/mapping algorithms.
- Better quantify the "overlap" of memory utilization for AMR calculations.
    - Dynamic nature of
- Addressed major inefficiencies on MCR & ALC:
    - Loop over Gather's replaced by MPIAlltoall (4x speedup for single level, 4x speedup for multiple level)

Applications on BG/L

- What's this in real units (M=1.3, air/SF6)?
  - Taylor length , λ, is O(50 μm) with Re = 2000 → ~ 2 cells/ λ on BG/L.

  - Most highly resolved 2D shock-driven flow done to date, e.g. M=1.2 shock/jet interaction used 20 cells/ λ.

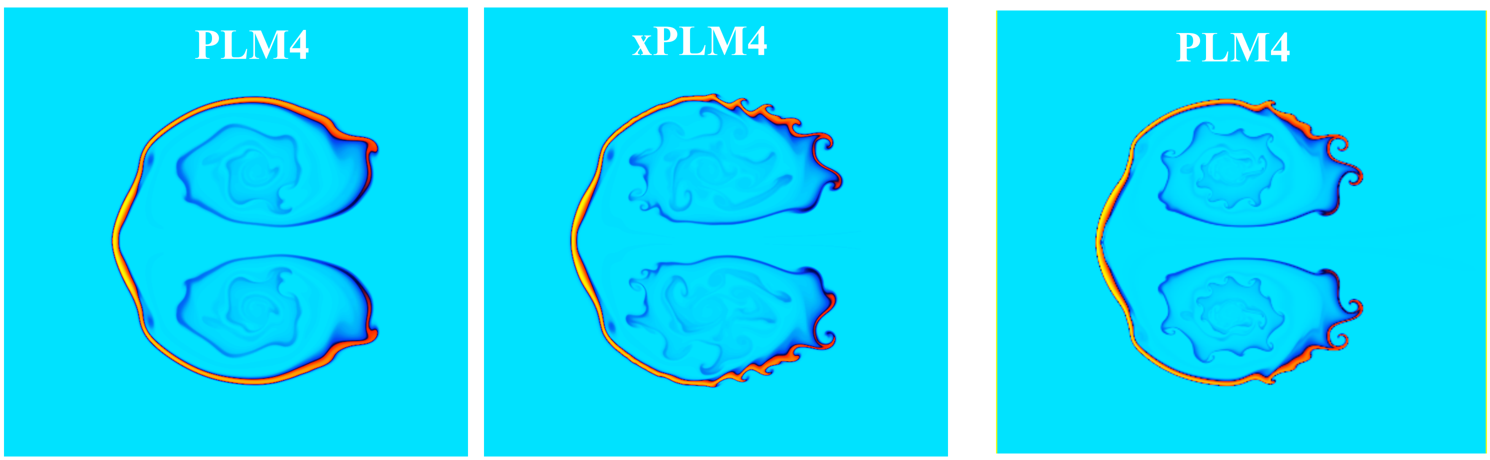


Applications on BG/L

- Improvements to the underlying Godunov method provides additional effective resolution

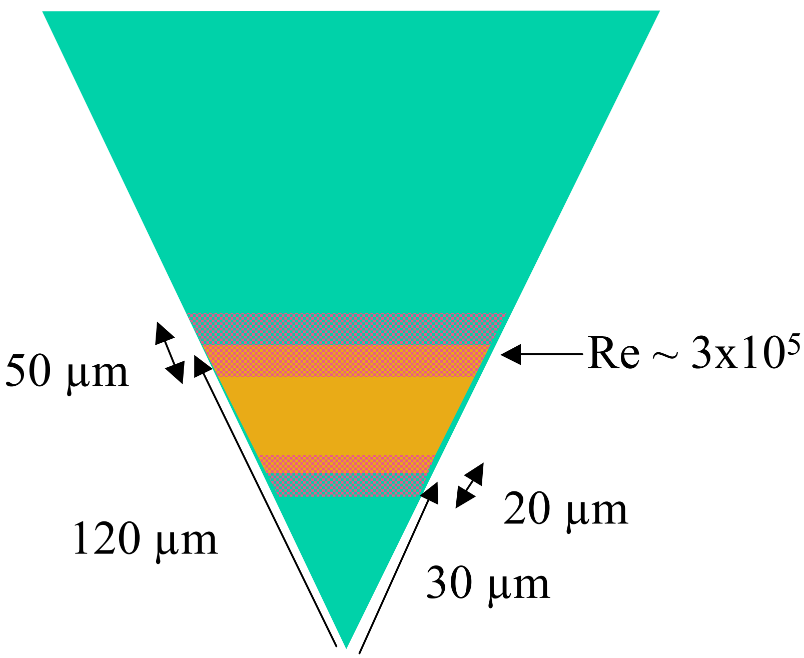
    - Increased “roll-up” indicates improved resolving power (reduced intrinsic dissipation).

Δx = 31.25 micron Δx = 1.95 micron



Applications on BG/L

- Examine turbulence in a NIF double shell ignition capsule
  - 90 degree 3D wedge with cell size ~ λ (Taylor Length)
  - Just before ignition, Re ~ 3×10^5 at outer mix region → λ ~ 0.1 µm
    - Assuming Δx ~ λ, gives ~1600^3 effective problem size



- Requires scalable and efficient linear solvers for the nonlinear problem

- Memory footprint for linear solver appears to be within the footprint for full-memory hydro.

- Requires Export Controlled database (EOS and opacity).

Summary/Strategy for Success

- Target high-resolution prototype RM problem
  - BG/L capability should allow realization of a fully-resolved shock-driven turbulent flow
    - Late time single-shock asymptotic state
    - Late time multiply-accelerated turbulent state
- Target high-resolution prototype of a two-shell NIF ignition capsule (ditto above)

Summary/Strategy for Success

- Utilize the BG/L Simulator on ALC
  - Single level tests (multiple grids) to test performance and identify issues
  - Adaptive tests (multiple levels, multiple grids) to test performance and identify issues.
  - Scrutinize the linear solvers and their usage/performance in Raptor.
- Additional development
  - Re-work our I/O model, grid distribution algorithm (hardware awareness)
  - Convert some post-processed diagnostics to runtime ones
  - Additional items as they appear.